The number of tree stars is $O^*(1.357^k)$

Bernhard Fuchs [a] Walter Kern [b,*] Xinhui Wang [b,1]

[a] *University of Cologne, Center for Applied Computer Science Cologne, Group AFS, Weyertal 80, 50931 Köln, Germany*

[b] *University of Twente, Department of Applied Mathematics, Faculty of EEMCS, P.O.Box 217, 7500 AE Enschede, the Netherlands*

Abstract

Every rectilinear Steiner tree problem admits an optimal tree T^* which is composed of *tree stars*. Moreover, the currently fastest algorithms for the rectilinear Steiner tree problem proceed by composing an optimum tree T^* from tree star components in the cheapest way. The efficiency of such algorithms depends heavily on the number of tree stars (candidate components). Fößmeier and Kaufmann [9] showed that any problem instance with k terminals has a number of tree stars in between 1.32^k and 1.38^k (modulo polynomial factors) in the worst case. We determine the exact bound $O^*(\rho^k)$ where $\rho \approx 1.357$ and mention some consequences of this result.

Key words: Rectilinear Steiner tree, Terminal points, Tree star

1 Introduction

Given a weighted graph (V, E) on $n = |V|$ nodes, non-negative edge weights $c : E \to \mathbb{R}_+$, a set $Y \subseteq V$ of k *terminal nodes* (or *terminals*, for short), the *Steiner tree problem* asks for exhibiting a shortest (*i.e.*, min cost) subtree $T^* = T^*(Y)$ of (V, E) spanning all terminals.

The most well-known algorithm for solving Steiner tree problems is the so-called Dreyfus-Wagner [1] algorithm, a certain dynamic programming ap-

* Corresponding author.
Email addresses: `bfuchs@zpr.uni-koeln.de` (Bernhard Fuchs), `kern@math.utwente.nl` (Walter Kern), `xinhuiw@math.utwente.nl` (Xinhui Wang).

1 Supported by Netherlands Organization for Scientific Research (NWO) grant 613.000.322 (Exact Algorithms).

proach that computes an optimum tree T^* in time $O^*(3^k)$. Here and in what follows, we use the O^*-notation to indicate that factors of order $O(poly(n))$ are suppressed. (In the rectilinear case we study here, $n = O(k^2)$, so equivalently, we suppress factors of order $O(poly(k))$.) The currently fastest algorithm, due to [11] resolves the problem in $O^*((2+\epsilon)^k)$ for any $\epsilon > 0$. We admit, however, that the result is purely theoretical and the algorithm is not expected to be of any use in practice.

The most interesting problems in practice are actually so-called *rectilinear* problems, where the terminal set is a finite set $Y \subseteq \mathbb{R}^2$ and the underlying graph (V, E) is the so-called *Hanan grid*: If $X_1 \subseteq \mathbb{R}$ resp. $X_2 \subseteq \mathbb{R}$ denote the projections of Y onto the first respective second coordinates, then $V = X_1 \times X_2$ and E is the complete set of edges $e = (u, v)$ with l_1-metric $c(e) = ||u - v||_1$.

In general (due to the non-negativity of the edge costs) every leaf of $T^* = T^*(Y)$ is necessarily a terminal. In addition, T^* may contain some terminals in its interior. These *interior terminals* split T^* into *components* (subtrees). In the rectilinear case, a lot is known about the structure of such components (*cf.* below and Section 2).

For simplicity, let us assume that the given instance $Y \subseteq \mathbb{R}^2$ consists of k points with pairwise different first resp. second coordinates, so that the associated Hanan grid has exactly $n = k^2$ nodes. This may always be achieved by perturbation. For example, if the original instance is defined by $Y = \{y_1, \cdots, y_k\} \subseteq \mathbb{Z}^2$, we may chose $\epsilon_1 > \cdots > \epsilon_k > 0$ sufficiently small and replace each y_i by

$$\widetilde{y_i} := y_i + (\epsilon_i, \epsilon_i^2).$$

The resulting set $\widetilde{Y} \subseteq \mathbb{R}^2$ has pairwise different coordinates. Moreover, if $\sum \epsilon_i + \epsilon_i^2 < \frac{1}{2}$, any optimum Steiner tree $\widetilde{T}^*$ for $\widetilde{Y}$ must correspond to an optimum Steiner tree T^* for Y. Note that, in addition, the perturbed instance $\widetilde{Y}$can be assumed to generate a Hanan grid without any induced squares. (Choose $\epsilon_1, \cdots, \epsilon_k$ so as to ensure that $\epsilon_i - \epsilon_j \neq \epsilon_h^2 - \epsilon_l^2$ holds for all pairwise different i, j, h, l).

In what follows we will assume throughout that $Y \subseteq \mathbb{R}^2$ is perturbed in this way. A well known result of Hwang ([2],[10]) then states the existence of an optimum Steiner tree $T^* = T^*(Y)$ with each component of the following form (*Hwang topology*): There are two special terminals, the *root* r and the *tip* t of the component, connected to each other by a horizontal and vertical line segment (the two *legs* of the component). These two legs are incident in a common endpoint $c \subseteq \mathbb{R}^2$, the *corner* of the component. The leg $[r, c]$ is called the *long leg* or (*Steiner*) *chain*, the other leg $[t, c]$ is called the *short leg* of the component. The chain has an arbitrary number of straight line segments attached to it from both sides alternatingly, each connecting exactly one terminal to the chain. In addition, there may be one exceptional terminal connected to the short leg (*cf.* Fig.1). The degree three nodes of the component (*i.e.* all interior nodes except the corner) are called *Steiner nodes*. We usually draw the Steiner chain horizontally in the direction of the positive x-axis as in Fig.1 below. The terminals $y \neq r, t$ that are attached to the chain from above resp. below are referred to as *upper* resp. *lower* terminals. The optional additional terminal attached to the short leg will not be of much interest for our purposes.

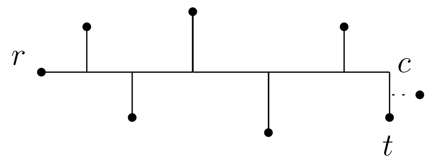

Fig. 1. A Hwang tree with optimal exceptional terminal

In what follows, a Steiner tree (component) with Hwang topology as above will be simply called a *Hwang tree.* A *Hwang set* is a set $X \subseteq Y$ which is the terminal set of at least one Hwang tree. We let $H(X)$ denote the shortest Hwang tree for X. By slightly misusing the notation, we also interpret $T^*(X)$ and $H(X)$ as the length of an optimum Steiner tree resp. Hwang tree for X. In case $X \subseteq Y$ is not a Hwang set, we define $H(X) = \infty$.

In the literature, Hwang sets/trees are also known as *full sets* and *full components*, as Hwang trees are candidates for T^*-components. Ganley and Cohoon ([3]) present a straightforward dynamic program computing an optimum Steiner tree T^* by composing T^* from Hwang trees in the cheapest way:

- Compute $H(X_1)$ for all $X_1 \subseteq Y$.
- Compute recursively for all $X \subseteq Y$

$$T^*(X) \ := \ \min_{X = X_0 \bowtie X_1} T^*(X_0) \ \cup \ H(X_1),$$

where

$$X = X_0 \bowtie X_1 \ \Leftrightarrow \ X = X_0 \cup X_1, \text{ and } |X_0 \cap X_1| = 1.$$

In [3], it is shown that there are (modulo polynomial factors) at most 1.62^k Hwang sets $X_1 \subseteq Y$. More generally, every $X \subseteq Y$ of size $i \leq k$ has at most 1.62^i Hwang subsets $X_1 \subseteq X$. So the above dynamic program has a running time of order

$$O^*(\sum_{i=1}^{k} \binom{k}{i} 1.62^i) = O^*(2.62^k).$$

Fößmeier and Kaufmann ([9]) further restrict the set of candidates for T^*-components by showing that each T^*-component can be assumed to be a so-called *tree star* (a Hwang tree with certain additional properties, *cf.* Section

2). They show that, in the worst case, the number of tree stars is in between 1.32^k and 1.38^k, yielding an improvement of the running time in the above dynamic program to $O^*(2.38^k)$. The analysis leading to the upper bound in [9] is rather involved (16 pages). We present a somewhat simpler approach yielding a tight bound of $O^*(1.357^k)$.

The currently fastest algorithms in practice ([7],[10]) first compute Hwang trees for all candidate sets and then seek to compose the optimum tree from these candidate sets - not necessarily by dynamic programming, but rather by solving a related integer program. In any case, the number of candidate sets determined in the preprocessing phase is crucial for the efficiency of the algorithm. Having a tight bound on the number of tree stars also allows us to estimate the impact of possible further restrictions on the candidate sets. For example, [8] exhibits additional properties of T^*-components (*cf.* Section 2), which are both natural and helpful in practice. As it turns out, however, the number of tree stars with these additional properties is still $O^*(1.357^k)$ in the worst case. So from a theoretical point of view, the new properties are of no help.

2 Tree stars

Consider an optimal Steiner tree T^* for a (suitably perturbed) instance $Y \in \mathbb{R}^2$. According to Hwang's theorem, we may assume that each component of T^* is a Hwang tree with root $r \in Y$, tip $t \in Y$ and terminals, say, $y_0, \cdots, y_p$ attached to the chain as in Fig.2. (with possibly an additional terminal y_{p+1} joined to the short leg).

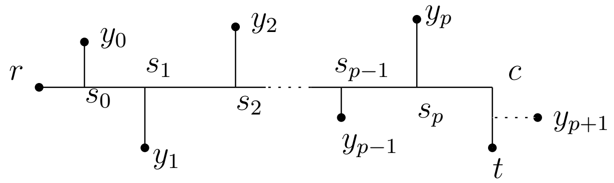

Fig. 2. Labelling the nodes in a component.

We denote by $s_0, \cdots, s_p$ the corresponding *Steiner points i.e.*, the degree 3 nodes which are the projections of the y_i's ($i = 0, \cdots, p$) onto the chain. (Clearly, in case there is an additional terminal y_{p+1}, we also have an additional Steiner point s_{p+1}. In what follows, however, we restrict our attention to $y_0, \cdots, y_p$ and $s_0, \cdots, s_p$ so that we do not have to distinguish between different types of Hwang trees.)

Let $S \subseteq \mathbb{R}^2$ denote the set of Steiner nodes of T^*. Clearly, being an optimal Steiner tree, T^* must be an MST for the set $Y \cup S$. This simple necessary condition on T^* in turn implies certain properties of the components of T^*. We present some of these properties (*empty regions conditions*, *cf.*, *e.g.*, [8]) below, including the simple proofs for convenience.

A *diamond* is a square with diagonal $[y_i, s_i]$, $i = 0, \cdots, p$ or $[s_i, s_{i+1}]$, $i = 0, \cdots, p-1$, or $[r, s_0]$. We then observe that $T^* = MST(Y \cup S)$ implies that every component of T^* as in Fig.3 must have *empty diamonds* in the sense that the interior of each diamond may not contain any terminal $y \in Y$ (*cf.* Fig.3):

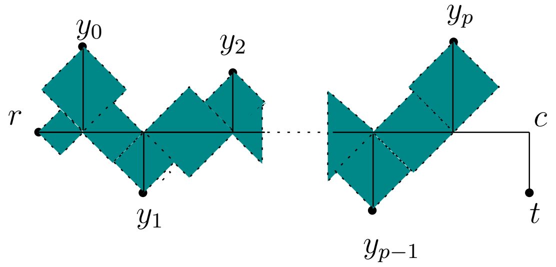

Fig. 3. Diamonds must be empty.

Lemma 2.1. Diamonds must be empty.

Proof. Assume to the contrary, that for some component, say, a diamond $D = D[y_i, s_i]$ with diagonal $[y_i, s_i]$ contains a terminal $y \in Y$ in its interior. Let y' denote the projection of y onto $[y_i, s_i]$. Adding, $e := [y, y']$ to T^* closes a circuit that contains either $f := [y', y_i]$ or $f := [y', s_i]$. In both cases, $T^* \backslash f \cup e$ would be a shorter tree, a contradiction.

Emptiness of diamonds of the form $D = D[s_i, s_{i+1}]$ follows in the same way.

■

Next we consider *rectangles* $R = R[y_i, s_{i+1}]$ or $R = R[y_i, s_{i-1}]$, defined by their diagonal $[y_i, s_{i+1}]$ resp. $[y_i, s_{i-1}]$, $i = 1, \cdots, p-1$ (*cf.* figure 4). Again, rectangles must be empty regions in the above sense.

Lemma 2.2. Rectangles must be empty.

Proof. Assume to the contrary that, say, some component of T^* contains a nonempty rectangle, say, $R = R[y_i, s_{i+1}]$. So R contains some $y \in Y$ in its interior. The rectangle R has sides $e = [y_i, s_i]$ and $f = [s_i, s_{i+1}]$. Let y_e and y_f denote the projections of y onto e resp. f. Recall from Section 1 that we may assume w.l.o.g. that the Hanan grid generated by Y does not contain any squares.

Thus we may assume w.l.o.g. that, say, y is closer to e than to f. Removing the segment $[y_e, s_i]$ from T^* splits T^* into two subtrees T_1 and T_2 containing

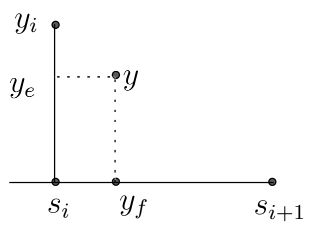

Fig. 4. A nonempty rectangle.

y_e resp. y_f. If $y \in T_2$, then $T_1 \cup T_2 \cup [y, y_e]$ is shorter than T^*, a contradiction. Hence $y \in T_1$ must hold. But then $T_1 \backslash [y_i, y_e] \cup T_2 \cup [y, y_f]$ is shorter than T^*. ■

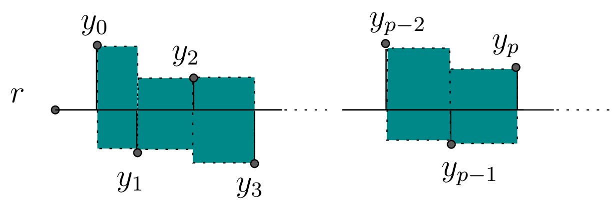

Fig. 5. Shaded regions must be empty.

The empty rectangles condition is rather restrictive: The number of Hwang trees satisfying the empty rectangles condition is $O^*(1.42^k)$, *cf.*[9] (as compared to $O^*(1.62^k)$ without this restriction, *cf.* [3]). This can be seen as follows. Any two consecutive terminals y_i and y_{i+2} "above" the chain uniquely determine the terminal y_{i+1} in between them on the opposite side of the chain. (Namely the one that is closest to the chain). This leads to a bound of $O^*(2^{\frac{p}{2}}) = O^*(2^{\frac{k}{2}}) = O^*(1.42^k)$ for the number of such Hwang trees in a straightforward way.

A third empty regions condition (*cf.* [4], [5], [8]) is as follows. Let $B_d(x)$ and $B_d(C)$ denote the l_1-balls of radius $d > 0$ around $x \in \mathbb{R}^2$ resp. the Steiner chain C. Let $d_i > 0$ denote the distance of y_i to C. For $i = 0, \cdots, p$, we let

$$\Delta(y_i) := B_{d_i}(y_i) \cap B_{d_i}(C)$$

denote the *triangle* defined by y_i. The following result strengthens the empty diamond condition for diamonds with diagonal $[y_i, s_i]$ (*cf.* Fig.6):

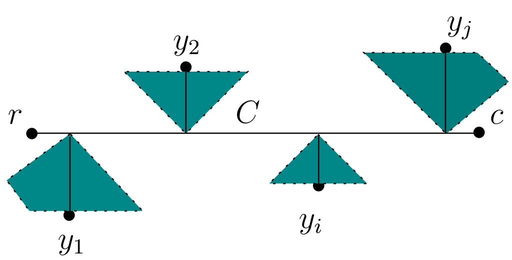

Fig. 6. Shaded region must be empty.

Lemma 2.3. ([8]) Each triangle $\Delta(y_i), i = 0, \cdots, p$ must be empty.

Proof. The proof is similar to the one of Lemma 2.2. Assume $y \in Y$ is in the interior of $\Delta(y_i)$. Removing $[y_i, s_i]$ from T^* would leave two subtrees T_1 and T_2 containing y_i resp. s_i (and hence C). If $y \in T_2$ then joining y to y_i would yield a tree shorter than T^*, a contradiction. Similary, $y \in T_1$ would imply a shorter tree, obtained by joining y to C. ■

In [9] (*cf.* also [6]), a *tree star* is defined to be a Hwang tree that satisfies the empty diamonds and empty rectangles condition and is, in addition, an MST of its terminals and Steiner points. So in particular, the part of the tree induced by $r_1, s_0, \cdots, s_p$ and $y_0, \cdots, y_p$ must be an MST of these points. The latter is equivalent to the following *weak empty triangle condition*: No $\Delta(y_i)$ must contain any y_j $(j = 1, \cdots, p)$ in its interior.

A fourth empty regions condition is discovered in [8]: For $i = 1, \cdots, p-1$, let

$$r_i =: \min\{||s_i - s_{i-1}||, ||s_{i+1} - s_i||, ||y_i - s_i||\}.$$

The *empty circles* condition states that each $B_{r_i}(s_i)$ must be empty, *i.e.*, contain no terminals in its interior, *cf.* Fig.7. The proof is left to the reader.

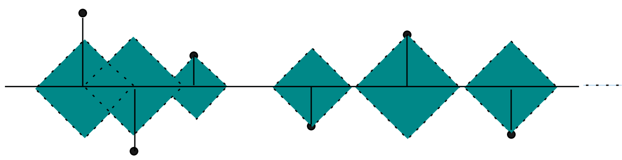

Fig. 7. Shaded regions must be empty.

Föẞmeier and Kaufmann [9] present a rather involved analysis showing that the number of tree stars is bounded by $O^*(1.38^k)$ and provide an example problem allowing 1.32^k tree stars. In Section 3, we take a somewhat simpler approach, leading to a bound of $O^*(1.357^k)$. Section 4 provides an example proving that our bound is tight.

3 The upper bound

Let $\alpha \approx 1.8393$ denote the unique real root of the polynomial $x^3 - x^2 - x - 1$. Our main result can then be stated as:

Theorem 3.1 The number of tree stars is bounded by $O^*(\sqrt{\alpha}^k) \approx O^*(1.357^k)$.

To prove Theorem 3.1 we consider a (fixed) Steiner chain C with terminals $a_0, \cdots, a_{l+1}$ above and $a'_0, \cdots, a'_l$ below the chain, so that a'_i is in between a_i

and a_{i+1}. We let z_i resp. z'_i denote the corresponding potential Steiner points, *cf.* Fig.8. We seek to analyze the number of tree stars that have C as Steiner chain, and a_0 and a_{l+1} as first resp. last upper terminal attached to C.

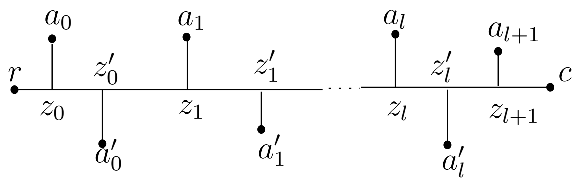

Fig. 8. Terminals above and below the chain.

In what follows, a *tree star* will always mean a tree star with chain C and a_0, a_{l+1} as first resp. last upper terminal. We are interested in which of the remaining terminals a_i $(i = 1, \cdots, l)$ such a tree star may include. Slightly misusing our notation, we treat each a_i also as a boolean variable indicating whether a_i is included in a given tree star or not. So we define a *tree star sequence* (TSS) to be sequence $a_0, \cdots, a_{l+1} \in \{0,1\}^{l+2}$ that corresponds to a tree star as above (hence, in particular, $a_0 = a_{l+1} = 1$). To prove Theorem 3.1, it suffices to show that the number q_l of TSS's is bounded by $O^*(\alpha^l)$. (Note that $l \leq k/2$ must hold.)

We start by providing various constraints on tree star sequences. For example, if $a_j \in \Delta(a_i)$ (*cf.* Fig.9), then the weak empty triangles condition implies that $a_i = a_j = 1$ cannot occur in a TSS. We say that 11 is *forbidden* for $a_i a_j$ in this case.

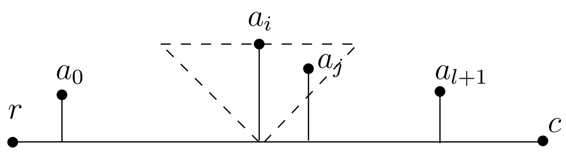

Fig. 9. 11 is forbidden for $a_i a_j$.

Another similar type of constraint is presented in Lemma 3.1 below. Let d_i and d'_i denote the distances of a_i resp. a'_i from C.

Lemma 3.1 If $d_i > d_{i+1}$ and $d'_i > d'_{i+1}$, then 10 is forbidden for $a_i a_{i+1}$ (*cf.* Fig.10).

Proof. Assume to the contrary that some TSS has $a_i = 1$ and $a_{i+1} = 0$. Let $j > i+1$ be the first index with $a_j = 1$. The lower terminal a'_k to be included in the corresponding tree star in between a_i and a_j is then at least as close to the chain as a'_{i+1} (according to the empty rectangles condition, a'_k is the lower terminal in between a_i and a_j which is closest to the chain). So $k \geq i+1$, contradicting the empty rectangle condition (as a_{i+1} is contained in the rectangle $R[a_i, z'_{i+1}]$ with diagonal $[a_i, z'_{i+1}]$). ■

Lemma 3.1 shows that any *local minimum* a_i, *i.e.*, any a_i with $d_{i-1} > d_i <$

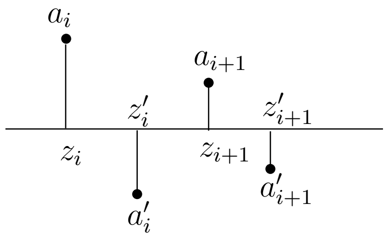

Fig. 10. 10 is forbidden for $a_i a_{i+1}$.

d_{i+1} implies a forbidden 10 for $a_{i-1}a_i$ or a forbidden 01 for $a_i a_{i+1}$ (depending on whether $d'_{i-1} > d'_i$ or $d'_{i-1} < d'_i$ holds). In some cases, we can derive an additional constraint:

Lemma 3.2. Let a_i be a local minimum with, say, $d'_{i-1} > d'_i$. If $d_{i-2} > d_i$, then either 100 is forbidden for $a_{i-2}a_{i-1}a_i$ or 001 is forbidden for $a_{i-1}a_i a_{i+1}$, depending on whether $d'_{i-2} > d'_i$ or not.

Proof. Consider first the case where $d'_{i-2} > d'_i$ (*cf.* figure 11) and assume to the contrary that $a_{i-2}a_{i-1}a_i = 100$ is part of a TSS, *i.e.*, there is a tree star T that includes a_{i-2}, but neither a_{i-1} nor a_i as upper terminal. Let $a_j, j > i$, denote the first upper terminal included in T. The corresponding lower terminal in between a_i and a_j is then either a'_i (as $d'_i < d'_{i-1}$ and $d'_i < d'_{i-2}$) or some lower terminal $a'_r, \;\; r \geq i$. Then $a_i \in R[a_{i-2}, z'_r]$ violates the rectangle condition.

The case where $d'_{i-2} < d'_i$ is similar. ■

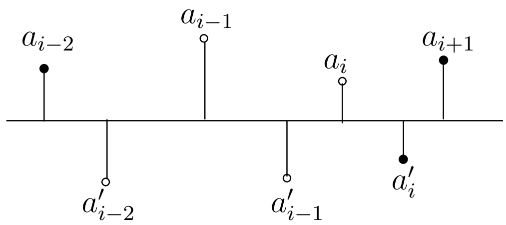

Fig. 11. Illustration of Lemma 3.2.

Lemma 3.3. Assume $d_{i-1} > d_i > d_{i+1}$ and $d'_{i-1} < d'_i < d'_{i+1}$ holds for some $1 \leq i \leq l{-}1$. Then either $a_i \in \Delta(a_{i-1})$ or $a'_i \in \Delta(a'_{i+1})$ or $\{a_i, a'_i\} \cap D[z_{i-1}, z_{i+1}] \neq \emptyset$.

Proof. Assume to the contrary that neither of these three possibilities occurs. Then (*cf.* Fig.12) a_i must be to the right of a'_i, which is ridiculous. ■

This simple observation leads to the following constraints on TSS's:

Lemma 3.4 Assume $d_{l-2} > \cdots > d_{l+1}$ and $d'_{l-3} < \cdots < d'_l$. Then either of the following holds:

a) 11 is forbidden for $a_{l-2}a_{l-1}$.
b) 11 is forbidden for $a_{l-1}a_l$.

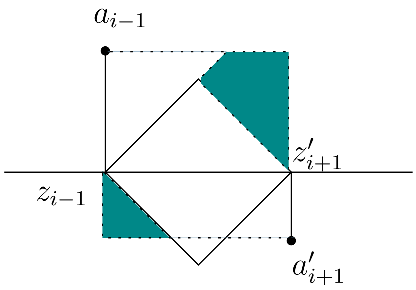

Fig. 12. Feasible regions for a_i resp. a'_i.

c) 000 is forbidden for $a_{l-2}a_{l-1}a_l$.

Proof. If $a_{l-1} \in \Delta(a_{l-2})$, then a tree star may not contain both a_{l-1} and a_{l-2} (weak empty triangles condition), so 11 is forbidden for $a_{l-2}a_{l-1}$. Similarly, if $a'_{l-1} \in \Delta(a'_l)$, a tree star may not contain both a'_{l-1} and a'_l. Consequently, it must not contain both a_{l-1} and a_l (because together with a_{l+1}, these would imply the inclusion of the lower terminals a'_{l-1} and a'_l). Thus $a'_{l-1} \in \Delta(a'_l)$ forbids 11 for $a_{l-1}a_l$.

According to Lemma 3.3, we are left to analyze the case where $\{a_{l-1},\ a'_{l-1}\} \cap D[z_{l-2}, z_l] \neq \emptyset$. We claim that 000 is forbidden for $a_{l-2}a_{l-1}a_l$ in this case. Indeed, assume to the contrary that T is a tree star that does not include any of a_{l-2}, a_{l-1} and a_l. Due to our assumptions $d_{l-2} > \cdots > d_{l+1}$ and $d'_{l-3} < \cdots < d'_l$, we conclude that T does not contain any of $a'_{l-2},\ a'_{l-1}$ and a'_l either. Let a'_i denote the last lower terminal contained in T. Hence $i \leq l-3$. Then $D[z'_i, z_{l+1}] \supseteq D[z_{l-2}, z_l]$ is nonempty, contradicting the empty diamonds condition. ■

We are now prepared to prove our main result in a special (though crucial) case:

Lemma 3.5 Let $d_0 > \cdots > d_{l+1}$ and $d'_0 < \cdots < d'_l$. Then $q_l \leq 1.183\alpha^l$.

Proof. For $l \leq 2$ the claim is trivial. (Indeed, $q_2 \leq 2^2 \leq 1.183\alpha^2$.) Hence assume $l \geq 3$. First note that, due to the special structure of our instance (distances d_j decreasing, and d'_j increasing), a tree star T which does not include a_i, also does not include a'_i. The number of TSS with $a_i = 0$ is therefore at most q_{l-1} by induction. (It might actually be less in case some TSS for the instance with a_i and a'_i removed corresponds to a Hwang tree containing a_i or a'_i in a forbidden region.)

We proceed by induction on l. According to Lemma 3.4, there are three possible cases:

c) 000 is forbidden for $a_{l-2}a_{l-1}a_l$. Induction then yields

$$q_l \leq 1.183 \cdot [\alpha^{l-1} + \alpha^{l-2} + \alpha^{l-3}] = 1.183\alpha^l,$$

where the terms in brackets account for the TSS's ending with $11, 101$ and 1001, resp.

b) 11 is forbidden for $a_{l-1}a_l$. Induction gives

$$q_l \le 1.183 \cdot 3 \cdot \alpha^{l-2} \le 1.183\alpha^l,$$

where the term $3\alpha^{l-2}$ takes care of the TSS's ending with $001, 011$ and 101.

a) 11 is forbidden for $a_{l-2}a_{l-1}$. Induction yields

$$q_l \le 1.183 \cdot 6 \cdot \alpha^{l-3} \le 1.183\alpha^l,$$

where the term $6\alpha^{l-3}$ accounts for the 6 possible endings of TSS's $00*1, 01*1$ and $10*1$. ■

The second assumption in Lemma 3.5 can be easily removed:

Lemma 3.6 Assume $d_0 > \cdots > d_{l+1}$. Then $q_l \le 1.183\alpha^l$.

Proof. The case where $d'_0 < \cdots < d'_l$ is settled by Lemma 3.5. Hence assume that $d'_i > d'_{i+1}$ for some i. Then Lemma 3.1 applies, showing that 10 is forbidden for a_ia_{i+1}. Thus induction gives

$$q_l \le 1.183 \cdot \alpha^{l-1} + 1.183^2 \cdot \alpha^{l-2} \le 1.183\alpha^l.$$

Here, the term $1.183\alpha^{l-1}$ accounts for the TSS's with $a_i = 0$ and the term $1.183^2 \cdot \alpha^{l-2}$ upper bounds the number of TSS's, with $a_ia_{i+1} = 11$. ■

Lemma 3.7 Let $d_0 < \cdots < d_j > d_{j+1} > \cdots > d_{l+1}$. Then $q_l \le 1.4\alpha^l$.

Proof. For $l \le 4$ the claim is trivial (as $2^4 < 1.4\alpha^4$). Hence assume $l \ge 5$. If $d'_i > d'_{i+1}$ for some $i \ge j$, then 10 is forbidden for a_ia_{i+1}. Thus induction yields

$$q_l \le 1.4 \cdot \alpha^{l-1} + 1.4 \cdot 1.183\alpha^{l-2},$$

where the first term accounts for all TSS's with $a_i = 0$ and the second term accounts for all TSS's with $a_i = 1$ and $a_{i+1} = 1$. (Observe that Lemma 3.6 applies to the subsequence $a_{i+1}, \cdots, a_{l+1}$.) We conclude that $q_l \le 1.4\alpha^l$ in this case. Similarly, the claim follows in case $d'_i < d'_{i+1}$ for some $i < j$. (Consider the reverse sequence $a_{l+1}, \cdots, a_0$.)

Finally, assume that $d'_j < \cdots < d'_l$ and $d'_0 > \cdots > d'_{j-1}$ holds. We may then (by passing to the reverse order $a_{l+1} \cdots a_0$ if necessary) assume w.l.o.g. that $j \le l-3$ unless $l = 5$ and $j = 3$. In any case we may assume that $d_{l-2} > d_{l-1} > d_l > d_{l+1}$ and $d'_{l-3} < d'_{l-2} < d'_{l-1} < d'_l$, so that Lemma 3.4 applies. We distinguish between the three cases according to Lemma 3.4:

a) If 11 is forbidden for $a_{l-2}a_{l-1}$, induction yields

$$q_l \leq 1.4 \cdot 6 \cdot \alpha^{l-3} \leq 1.4\alpha^l,$$

where the term $6\alpha^{l-3}$ accounts for the TSS's ending with $00 * 1, 01 * 1$ and $10 * 1$.

b) If 11 is forbidden for $a_{l-1}a_l$, induction yields

$$q_l \leq 1.4 \cdot 3 \cdot \alpha^{l-2} \leq 1.4\alpha^l.$$

c) If 000 is forbidden for $a_{l-2}a_{l-1}a_l$, induction yields

$$q_l \leq 1.183[\alpha^{l-1} + \alpha^{l-2} + \alpha^{l-3}] = 1.4\alpha^l.$$

■

We now finally arrive at the

Proof of Theorem 3.1: We claim that $q_l \leq 1.4\alpha^l$ holds in general. We are left to deal with the case where some local minimum exists. Let a_i be the deepest local minimum, *i.e.*, d_i is minimal with the property that $d_{i-1} > d_i < d_{i+1}$. Assume w.l.o.g. that $d'_{i-1} > d'_i$. Then 10 is forbidden for $a_{i-1}a_i$. In case $i = 1$, we thus have $a_i = 1$ for every TSS (as $a_0 = 1$ is fixed), and the result follows by induction. Hence assume $i \geq 2$. We distinguish two cases:

1) $d_{i-2} > d_i$. In this case, either 100 is forbidden for $a_{i-2}a_{i-1}a_i$ or 001 is forbidden for $a_{i-1}a_ia_{i+1}$ (*cf.* Lemma 3.2).

In other words, $a_{i-1}a_i = 00$ either implies $a_{i-2} = 0$ or $a_{i+1} = 0$. In both cases we conclude by induction that the number of TSS's with $a_{i-1}a_i = 00$ is at most $1.4\alpha^{l-3}$. Hence, induction yields

$$q_l \leq 2 \cdot 1.4^2\alpha^{l-2} + 1.4\alpha^{l-3} \leq 1.4\alpha^l$$

(The first term bounds the number of TSS's with $a_{i-1}a_i = 01$ or 11.)

2) $d_{i-2} < d_i$. In this case $d_0 < d_1 < \cdots < d_{i-2} < d_{i-1}$ must hold (otherwise a_i were not the deepest local minimum). Bounding inductively the number of TSS's with $a_{i-1} = 0$ and $a_{i-1} = a_i = 1$, we get

$$q_l \leq 1.4 \cdot \alpha^{l-1} + 1.183 \cdot 1.4 \cdot \alpha^{l-2} \leq 1.4\alpha^l,$$

finishing the proof. ■

4 The lower bound

It is obvious from Lemma 3.5, what a worst case example matching the upper bound should look like. We let $z_0 = 0, z_1 = 1 - \epsilon$ and, in general, z_i is defined by

$$||z_i - z_{i-1}|| = (1-\epsilon)^2 ||z_{i-1} - z_{i-2}||,$$

for suitable $\epsilon > 0$. We let $d_0 = 1$ and $d_1 = (1-\epsilon)^2$, and in general,

$$d_{i+1} = (1-\epsilon)^2 d_i.$$

The lower terminals are given by $z'_l = l + \frac{1}{2}, z'_{l-1} = l - \frac{1}{2} + \epsilon, d_l = 1$ and

$$||z'_i - z'_{i-1}|| = (1-\epsilon)^2 ||z'_{i+1} - z'_i|| \text{ and } d'_i = (1-\epsilon)^2 d'_{i+1}.$$

For $\epsilon > 0$ sufficiently small, we have $z_i \approx i$ and $z'_i \approx i + \frac{1}{2}$. It is straightforward to check that any sequence $a_0, \cdots, a_{l+1}$ with no more than two consecutive zeroes is a TSS. Fig.13 below indicates some empty regions. (For simplicity, the figure is drawn with $\epsilon = 0$.) To verify, say, the empty diamonds condition, consider a diamond $D = D[z'_{i-2}, z_{i+1}]$ as in figure 13 below. A sequence with $a_{i-2}a_{i-1}a_ia_{i+1} = 1001$ would correspond to a tree T containing a_{i-2} and a_{i+1} on the upper side and a'_{i-2} on the lower side (as this is closest to the chain). For $\epsilon = 0$, the diamond D has no terminals in its interior, but *e.g.*, a_i is on its boundary, as $d_i = 1 = ||z_i - z_{i+1}||$ holds. For $\epsilon > 0$, we have

$$d_i = (1-\epsilon)^{2i} \quad \text{and} \quad \|z_i - z_{i+1}\| = (1-\epsilon) d_i,$$

so that a_i is not (no longer) contained in D. A symmetric argument applied to a'_{i-1} indeed shows that D is empty. Furthermore, any subtree fulfills the (weak) empty triangle condition. Hence indeed any sequence with no more than two consecutive zeroes is a TSS.

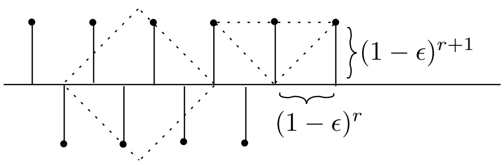

Fig. 13. A tight worst case example.

It is straightforward to check that the tree stars corresponding to such a TSS also satisfy the (strong) empty triangles and circles condition as mentioned in Section 2. (In addition to these empty regions conditions, [8] proves various upper bounds on the length d_{l+1} of the last vertical segment. To modify our worst case example so as to also meet these additional constraints, one simply has to choose the last terminal a_{l+1} sufficiently close to the chain.)

Summarizing, we conclude that

Proposition 4.1 In the worst case, there are up to $\Omega(\sqrt{\alpha}^k)$ tree stars satisfying the empty triangles and circles condition. In particular, the upper bound in Theorem 3.1 is tight.

5 Remarks and open problems

We like to remark that our upper bound of $O^*(1.357^k)$ can only be proved for suitably perturbed instances. Indeed, the worst case instance (Fig.13) in Section 4 with $\epsilon = 0$ would allow a lot more tree stars: Actually any sequence with at most 4 consective zeroes would be a TSS. (If $a_i = a_{i+5} = 1$ and $a_{i+1} = \cdots = a_{i+4} = 0$, the corresponding tree star must include a'_{i+2} as lower terminal.) This yields 1.96^l TSS's or 1.4^k tree stars (disregarding possible choices for the lower terminals).

A second point we want to stress is that what we count is the number of tree stars, rather than the actual number of potential components (candidate components) of the optimum tree T^*. For example, observe that none of the tree stars we count in our worst case example (Fig.13) in Section 4 occurs in the optimum tree. So it is quite possible that the number of "candidate sets" can be further reduced.

In this context it is of interest that (as proposed by one of the referees) we input our worst case example to GeoSteiner 3.1, a software package (*cf.* http://www.diku.dk/geosteiner/) which generates full components on the basis of empty regions conditions as well as other more "global" conditions. These other conditions (which are not known to us in detail) are seemingly quite strong, at least they ruled out most of our tree stars from the list of candidates so that the number of candidate sets generated for our worst case example was much less than α^l. In contrast, the "worst case example" from Fößmeier and Kaufmann ([9]) gave rise to many more candidate sets. The numbers of generated candidate sets for FK- instances and ours (labeled TS) for various values of l are shown in Table 1 below.

l	5	10	15	20	25
FK	104	235	448	817	1498
TS	29	54	79	104	129

Table 1 The number of candidate sets.

The experimental results from geosteiner seem to indicate that tree stars are not the final truth and that there are many more conditions on candidate sets that one should take into account. Yet, as mentioned earlier, knowing the exact number of tree stars may help us also to estimate more accurately the effect these additional conditions have on the number of candidate sets.

In practice, "most problem instances" produce an "almost linear" growth rate of the number of candidate sets (tree stars with empty triangles and empty circles) (*cf.* [8]). An intriguing open problem is whether one can exhibit conditions that imply a polynomial upper bound on the number of candidate sets. (This would imply a running time of $O^*(2^k)$ for the dynamic program in Section 1.) Another line of future research, as proposed by one of the referees, is to consider random instances in the spirit of [9].

6 Acknowledgements

We are grateful to an anouymous referee for many helpful suggestions and comments on an earlier version of this paper.

References

[1] S. E. Dreyfus and R. A. Wagner, (1972) The Steiner problem in graphs. *Networks*, 1, 195–207.

[2] F. K. Hwang, (1976) On Steiner Minimal Trees with Rectilinear Distance. *SIAM Journal on Applied Mathematics*, 30(1), 104–114.

[3] J. L. Ganley and J. P. Cohoon, (1994) Optimal rectilinear Steiner minimal trees in $O(n^2 2.62^n)$ time, in: *Proceedings of the Sixth Canadian Conference on Computational Geometry*, 308–313.

[4] J. S. Salowe and D. M. Warme, (1995) Thirty-Five Point Rectilinear Steiner Minimal Trees in a Day, *Networks*, 25, 69-87.

[5] J. S. Salowe and D. M. Warme, (1993) An exact rectilinear Steiner tree algorithm, *Proceedings of the International Conference on Computer Sesign*, 472-475.

[6] U. Fößmeier, M. Kaufmann and A. Zelikovsky, (1997) Faster Approximation Algorithms for the Rectilinear Steiner Problem, *Discrete and Computational Geometry*, 18, 93-109.

[7] D. M. Warme, (1998) PhD Dissertation, The University of Virginia, Department of Computer Science.

[8] M. Zachariasen, (1999) Rectilinear Full Steiner Tree Generation, *Networks*, 33, 125-143.

[9] U. Fößmeier and M. Kaufmann, (2000) On exact solutions for the rectilinear Steiner tree problem Part 1: Theoretical results. *Algorithmica*, 26, 68–99.

[10] M. Zachariasen, (2000) The Rectilinear Steiner Tree Problem: A Tutorial, in: *Steiner Trees in Industries*, X. Cheng and D.-Z. Du (eds.), Kluwer.

[11] B. Fuchs, W. Kern, D. Mölle, S. Richter, P. Rossmanith and X. Wang, (2006) Dynamic Programming for Minimum Steiner Trees, *Theory of Computing Systems*, to appear.